



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 11213029 A

(43) Date of publication of application: 06 . 08 . 99

(51) Int. Cl

G06F 17/50**H01L 21/82**

(21) Application number: 10018542

(71) Applicant: HITACHI LTD

(22) Date of filing: 30 . 01 . 98

(72) Inventor: SEKI MITSUO
ABE MASAHIRO

(54) DEVICE FOR EVALUATING PERFORMANCE OF REGISTER TRANSFER LEVEL LOGICAL DESCRIPTION HARDWARE

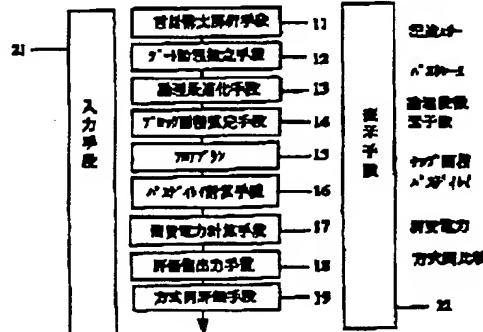
results to correspond to register transfer level logical description.

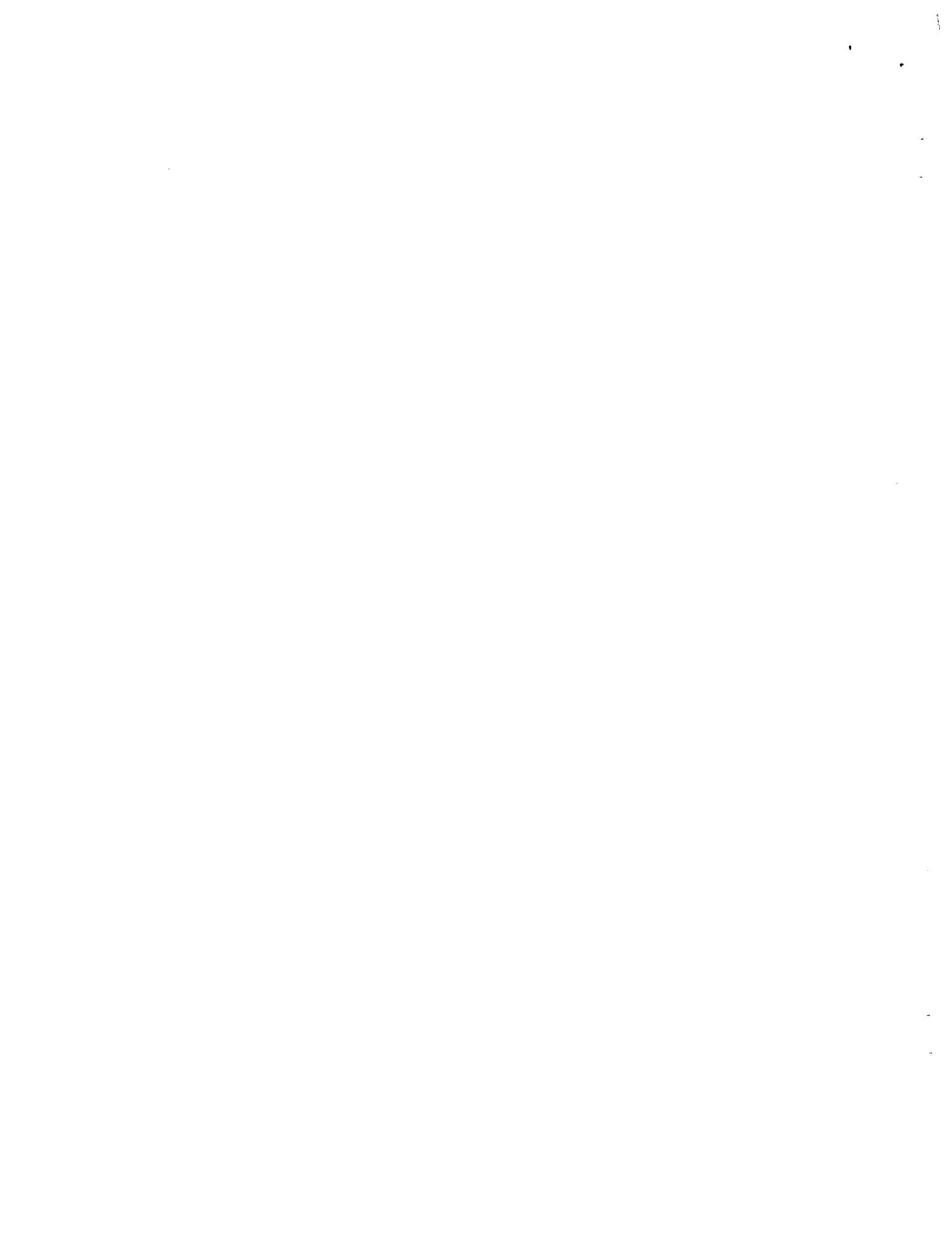
COPYRIGHT: (C)1998,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To synthetically evaluate three elements of an area, a delay and power consumption in register transfer level logical description without using synthesized gate level description or logical simulation, to execute logical design while considering the trade-off of hardware performance within short analytical time and to accurately grasp a problem on the register transfer level logical description by forming complete correspondence between a logical pass route and the description.

SOLUTION: An area is evaluated by a block area estimating means 14 and a floor plan 15, a delay is evaluated by the means 14, the plan 15 and a pass delay calculation means 16 and power consumption is evaluated by the means 14, the plan 15 and a power consumption calculation means 17. A gate logic estimating means 13 finds out initial logic based on language specification from logical description and allows the initial logic to correspond to all logical paths on the description independently of logically synthesized results. A display means 22 surely allows all evaluation





(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-213029

(43)公開日 平成11年(1999)8月6日

(51)Int.Cl.⁶

G 0 6 F 17/50
H 0 1 L 21/82

識別記号

F I

G 0 6 F 15/60

H 0 1 L 21/82

6 6 4 Z

6 6 6 A

6 6 8 A

C

審査請求 未請求 請求項の数13 ○L (全 15 頁)

(21)出願番号

特願平10-18542

(22)出願日

平成10年(1998)1月30日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 関 光穂

東京都小平市上水本町五丁目20番1号 株式会社日立製作所半導体事業部内

(72)発明者 安部 正秀

東京都小平市上水本町五丁目20番1号 株式会社日立製作所半導体事業部内

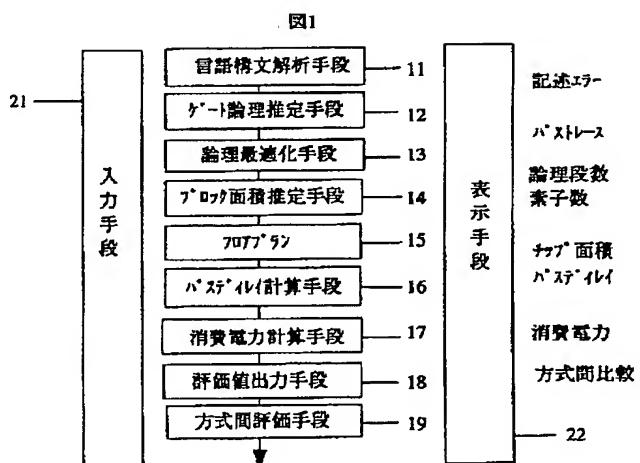
(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 レジスタ転送レベル論理記述ハードウェア性能評価装置

(57)【要約】 (修正有)

【課題】 レジスタ転送レベル論理記述に対して、論理合成後のゲートレベル記述や論理シミュレーションを用いて面積・ディレイ・消費電力の3つを総合的に評価し、短い解析時間でハードウェア性能のトレードオフを考えながら論理設計を行う。さらに論理パス経路とレジスタ転送レベル論理記述との対応を完全にとり記述上の問題点を正確に把握する。

【解決手段】 面積に関してはブロック面積推定手段14・フロアプラン15、ディレイに関しては面積に関する手段とバスディレイ計算手段16、消費電力に関しては面積に関する手段消費電力計算手段17により評価する。ゲート論理推定手段13により、論理記述から言語仕様に基づいた初期論理を求め、論理合成後の結果にかかわらず記述上の全ての論理パスへの対応を取る。また、表示手段22はすべての評価結果とレジスタ転送レベル論理記述との対応が必ず取れるようする。



【特許請求の範囲】

【請求項1】レジスタ転送レベル論理記述を入力してLSIハードウェア性能を予測評価する装置において、前記論理記述に基づき、各ブロックの大きさとブロック間の相対位置関係を決定するブロック面積推定手段と、前記ブロック間の相対位置関係情報に基づき、3次元的な容量を考慮したネットディレイを求めるバスディレイ計算手段と、消費電力を計算する消費電力計算手段と、複数のレジスタ転送レベル論理記述に対する上記各手段による評価値のトレードオフを表示する方式間評価手段とを備えたことを特徴とするレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項2】前記入力論理記述が複数階層を持つときに、下位階層の記述を先に解析して、該解析項目をライブラリへ入力して持ち、最後に最上位階層の解析を行う手段をさらに備えたことを特徴とする請求項1記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項3】前記入力した論理記述を言語仕様に基づいた初期論理に変換するゲート論理推定手段と、該入力記述上で解析可能な全ての論理バスの経路を表示する手段とをさらに備えたことを特徴とする請求項1記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項4】前記入力した論理記述を言語仕様に基づいた初期論理に変換した中に存在するブロックや素子に関して、ブロックや素子名、それが含まれる上位ブロック、論理段数、そのブロックに含まれる素子数、面積、消費電力、配置位置を表示する手段をさらに備えたことを特徴とする請求項1記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項5】前記請求項3記載の表示手段が、全ての論理バスに対してその始点・終点、バスディレイ、論理段数、バスの種類を表示することを特徴とする請求項3記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項6】前記請求項3記載の表示手段が、ひとつのバスが選択されると、そのバス上のネットと素子の接続関係を順に表示することを特徴とする請求項3記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項7】前記請求項3記載の表示手段が、ひとつのバスが選択されると、そのバス上の全てのネットと全ての素子に対応する記述部分をハイライトして表示することを特徴とする請求項3記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項8】前記請求項6記載の表示手段が、ネット名、そのネットが含まれるブロック名、そのブロックを含む上位ブロック、そのネットの出力素子名、そのネットのファンアウト数、そのネットの出力素子の論理段数、そのネットの入力素子名を表示することを特徴とする請求項6記載のレジスタ転送レベル論理記述ハードウ

エア性能評価装置。

【請求項9】前記請求項8記載の表示手段が、ネットの番号を選択すると、そのネットとそのネットの出力素子に対応する記述部分をハイライトして表示することを特徴とする請求項8記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項10】ブロック面積を推定し、そのブロックを配置し、また既作成のブロック情報を入力しフロアプランを行う手段をさらに備えたことを特徴とする請求項1記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項11】前記フロアプラン上で論理バス経路の表示をする手段をさらに備えたことを特徴とする請求項10記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項12】消費電力を計算した結果を時刻毎にグラフ表示する手段をさらに備えたことを特徴とする請求項1記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【請求項13】消費電力を計算するためのシミュレーションパターンを自動生成する手段をさらに備えたことを特徴とする請求項1記載のレジスタ転送レベル論理記述ハードウェア性能評価装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は半導体設計装置に係り、特にレジスタ転送レベル論理記述を入力してLSIのハードウェア性能評価を行う装置に関するものである。

【0002】

【従来の技術】ハードウェア設計言語と論理合成ツールが普及し、レジスタ転送レベルでLSI論理記述を行うことが標準的になったが、その論理記述からLSIがどのようなハードウェア性能になるかを予測評価するのは論理合成後のゲートレベル記述を用いて行うのが一般的である。論理合成には時間がかかること、ゲートレベル記述は膨大な量になること、より良いLSIを作るにはレジスタ転送レベル論理記述を良くする必要があること、といった点からレジスタ転送レベル論理記述から直接LSIハードウェア性能を予測評価する取組が行われるようになってきた。

【0003】LSIのハードウェア性能評価指標は主として、面積・ディレイ・消費電力の3つであるが、特開平06-83895では面積の予測、特開平05-174093ではディレイの予測、特開平07-230485では消費電力の予測を行っている。もうひとつの問題点として、論理合成はレジスタ転送レベル論理記述から変換した記述を最適化するので、論理合成結果とレジスタ転送レベル論理記述との対応が容易に取れないということがあったが、特開平05-242184ではその対応を取ろうとしている。

【0004】

【発明が解決しようとする課題】レジスタ転送レベル論理記述に対して面積・ディレイ・消費電力それぞれの評価指標を予測する発明はあるが、それら3つを総合的に評価するシステムがないという問題がある。3つのハードウェア性能を全て同時に最高にすることは難しく、それぞれのハードウェア性能のトレードオフを考えながら論理設計を行うことが必要であるため、ハードウェア性能評価システムは3つの指標を同時に解析できる必要がある。また、複数の記述に対してもそれぞれの解析結果を比較する必要がある。

【0005】評価精度の問題として、特開平06-83895ではハードウェア設計言語の各記述文の分類に対応したノウハウライブラリによって面積予測を行うので単純な和で計算される。ところが、論理合成は記述文の組み合わせによってより良い最適化を行うので、この方法では高速に予測はできるが精度が上がらないという問題がある。特開平05-174093ではごく簡易なフロアプランを行っているが、ディープサブミクロン世代の重要な問題である配線ディレイに関して容量や抵抗の考慮が3次元的になっていない。また、特開平07-230485の消費電力予測では論理合成をした後のゲートレベル記述論理を用い、かつ論理シミュレーションの結果を用いて計算を行っているが、論理合成やシミュレーションの時間がかかることおよびシミュレーションのためのテストパターン作成の工数を必要とするという問題がある。

【0006】特開平05-242184では論理合成結果とレジスタ転送レベル論理記述との対応を取って表示しようとしているが、合成結果から対応を取ろうとしているため合成結果にない情報は関連づけられないという問題があった。

【0007】

【課題を解決するための手段】3つの指標を同時に評価するために、面積評価に関してはブロック面積推定手段14・フロアプラン15、ディレイ評価に関してはブロック面積推定手段14・フロアプラン15・バスディレイ計算手段16、消費電力評価に関してはブロック面積推定手段14・フロアプラン15・消費電力計算手段17を持つ。ここでブロック面積推定手段14・フロアプラン15が共通にあるのはブロック面積とブロックの相対位置関係が配線容量と密接に関係し、ディレイと消費電力の評価値に影響するからである。

【0008】論理最適化手段13によって、ゲートレベル論理を入力せずに、高速にレジスタ転送レベル論理を最適化することによって推定精度の向上及び評価時間の短縮を行う。また、テストパターン作成手段80によって消費電力計算の人手作業時間を短縮する。

【0009】論理合成結果とレジスタ転送レベル論理記述との対応に関しては、ゲート論理推定手段13により、レジスタ転送レベル論理記述から言語仕様に基づいた初期論理に変換することによって、論理合成後の結果にか

かわらず全てのレジスタ転送レベル論理記述上の論理パス(論理の始点終点と処理の流れ)の対応を取る。

【0010】表示手段22はその他の手段で解析された結果を、それぞれの結果を表示するだけでなく、評価結果とレジスタ転送レベル論理記述との対応が必ず取れるように、また、評価指標間で対応が取れるように連動して表示する。

【0011】方式間評価手段19は、いろいろなレジスタ転送レベル論理記述に対して行った評価値を入力して、トレードオフをグラフで表示する。

【0012】ゲート論理推定手段12はレジスタ転送レベル論理記述から言語仕様に基づいた初期論理に変換するので、レジスタ転送レベル論理記述そのものの素直な解析結果を出力する。論理合成でもほぼ同じ初期論理から出発して論理の最適化を行うので、解析中の論理パスの始点終点が同一であれば、その途中の回路が論理合成によってどのように最適化されようとレジスタ転送レベル論理記述上にすべての対応を表示することができるよう作用する。

【0013】論理最適化手段13はゲート論理推定手段12により変換された初期論理を高速に最適化し、その後の指標の評価手段に最適化論理を与えることによって評価精度を高めると共に、消費電力計算をゲートレベル論理を入力して行なわなくともできるように作用する。

【0014】ブロック面積推定手段14とフロアプラン15は各ブロックの大きさとブロック間の相対位置関係を決定するので、微細なプロセスを使用したLSIで問題となるブロック内及びブロック間のネットのディレイを精度良く評価することができるように作用する。

【0015】バスディレイ計算手段16はブロック面積推定手段14とフロアプラン15によって求められた位置的な情報を元に3次元的な容量を考慮してネットディレイを求めると共に、ディレイのクリティカルな論理パスを抽出して問題点を示すように作用する。

【0016】テストパターン作成手段80は、消費電力計算のために論理シミュレーションを行うとき、そのテストパターンを作成する人手作業時間を大幅に短縮するので、時間を掛けて消費電力計算を行う手間を省く用に作用する。

【0017】消費電力計算手段17は、論理シミュレーション結果からの計算と入力ピンの信号変化情報のみでの計算と2通りの計算方法を備える。論理シミュレーション結果からの計算によって精度良い値を、入力ピンの信号変化情報のみでの計算では若干誤差は大きいものの高速に消費電力値を求められるように作用する。

【0018】表示手段22は解析された評価結果を、それぞれの結果だけでなく、評価結果とレジスタ転送レベル論理記述との対応が必ず取れるように、また、評価指標間で対応が取れるように連動して表示するので、レジスタ転送レベル論理記述上の問題点がただちに把握でき、

的にレジスタ転送レベル論理記述上の問題を解決できるように作用する。

【0019】方式間評価手段19は、より良い論理設計を模索するためにいろいろなレジスタ転送レベル論理記述を行ったときに、それらに対して行った評価値を入力してトレードオフをグラフで表示するので論理設計者の意思決定が迅速に行えるように作用する。

【0020】

【発明の実施の形態】図1から図27を用いて本発明の実施例を示す。図2は本発明のシステムの入出力ファイル構成を示す。入力するのは、レジスタ転送レベル論理記述(図中ではRTL、Register-Transfer Level 記述と表記)101、レジスタ転送レベル論理記述から推定したゲート論理を最適化する際に使用する論理素子群を格納する論理最適化用lib102、ネットディレイや面積を計算するときのLSIデバイス・プロセスのパラメータや評価精度を向上させるためのチューニングパラメータなどが格納されているデフォルトパラメータファイル103、そして出来合いの大型マクロセル(メモリやレジスタなど)の物理的・論理的情報であるLEFファイル104である。

【0021】入出力の両方を行うのは、本システムの解析結果や途中結果のメモリダンプを格納するdumpfile107、レジスタ転送レベル論理が階層的に記述される場合に下位階層のブロックの記述を先に解析した結果を格納しておくブロックlib106、フロアプラン結果からブロックの形状と配置位置を記述して論理合成システムにフィードバックするPDEFファイル105がある。dumpfile107を用いることによって、処理を途中で中断したり再開したりすることができるし、それを読み上げればいつでも詳細な解析結果をGUI画面から見ることができる。レジスタ転送レベル論理記述101が大規模で階層的に記述されている時は、一挙に全記述を読み上げて解析するのは計算機のメモリ量の問題から不可能であるため、下位階層について先に解析して、ブロックlib106に縮退させた情報で保存しておく、後に上位記述を読んで全体解析をする。

【0022】さらに複数のレジスタ転送レベル論理記述に対して方式間比較を行う場合には、各記述に対する解析値を各方式評価結果108として保存しておく、それを入力してグラフィカルに比較表示する。

【0023】つぎに本評価システムの処理の流れを示す。図1は本発明のシステムの処理の構成を示す図である。以下では図1中の各手段の詳細な構成と処理内容を詳細に述べる。

【0024】最初に入力手段21により図2中の入力ファイルを必要に応じて入力する。次に言語構文解析手段11がレジスタ転送レベル論理記述を入力する。言語構文解析手段11は図3に示すような手段から構成される。まずソース文字列入力手段31により記述を1行単位に入力する。そしてソース行管理手段32によりその行の文字列を

メモリ上に管理する。行を解析したとき、単語への分解手段33が実行されて行が解析され、単語に分解される。もし他の記述ファイルを参照しているときにはファイル管理手段34が実行されて、他の記述ファイルの内容を後に入力して解析を行えるようにする。分解された単語はその意味によって分類され、コード番号が付けられて、後のゲート論理推定手段12におけるコード番号に対応したゲート論理発生処理の識別となる。同時に単語位置管理手段35によってその単語がレジスタ転送レベル論理記述のどこに位置しているかを記憶する。これによって後にゲート論理推定手段12が発生するゲート論理素子や素子間を接続するネットがレジスタ転送レベル論理記述のどこに対応するか表示することができる。もし、ある単語が別の内容を持っているとき、例えばAAと書くと数字の10を意味すると言った表現がある場合には単語置換手段36がAAを10に置き換える処理を行う。最後に単語相関解析手段37が分類された単語間の相関解析を行う。これは例えば、ifに対してはelseがあったり、caseに対してはendcaseがあたりと、お互いに相関をなして初めて意味が完結するような表現があるからである。この相関解析で相関関係に矛盾が生じると記述エラーとしてエラーメッセージが outputされ、論理設計者に記述修正を促す。

【0025】次にゲート論理推定手段12が実行され、レジスタ転送レベル論理記述をそのまま素直に変換したゲート論理を生成する。まずネット宣言解析手段41が、レジスタ転送レベル論理記述中に宣言されているネット(そのブロック全体に対する入力信号や出力信号、及び途中の計算をつなぐ信号)の宣言に対して解析を行い使われる可能性のあるネットの候補リストを作る。そして、初期ネット作成手段42がそのブロック全体に対して入力となっているネットに対して入力からの接続情報を作る。次には言語構文解析手段11によって解析されている単語の分類と相関関係を用いて、ネット宣言を除く全ての単語について分類別に初期ゲート論理を発生していく。単語分類別素子発生手段43が論理素子を生成する。例えば記述中に=があればバッファを、if文があればその出力にセレクタ(実際にはAND, ORに分解)を生成する。そして単語素子間表示位置対応付け手段44が発生した素子とレジスタ転送レベル論理記述との対応を取る。例えばセレクタなら対応するifの場所を表示できるように対応情報を作る。次に発生させた素子間をネットで結ぶのであるが、記述中に明示されているネットを用いて結べる場合と新たにネットを作りだして結ぶ場合がある。例えばセレクタの場合はANDとORの2素子が使われるが、2素子の間のネットに関しては記述中に明示されていないので、非明示ネット作成手段45がネットを新たに生成し、前述のネット候補リストに加える。次にネット接続手段46がネット候補リストを参照して素子間を接続する。このネットに対しても単語ネット間表示位置対応付

け手段47がレジスタ転送レベル論理記述との対応を取る。ネットの接続が終わるとネット候補リスト更新手段48によって接続済みのネットは捨てられ、次の単語解釈により発生する素子との接続の候補リストに更新される。このようにしてレジスタ転送レベル論理記述中の全単語についての初期ゲート論理発生を行う。そして最後に外部接続ネット作成手段49がこのブロック全体に対する出力ネットを出力端子に接続する。これで必要な素子とその間を接続するネットの生成、素子やネットとレジスタ転送レベル論理記述との対応が作られたので、それらをスキャンして論理パスを見つける論理パス探索手段50が実行される。この手段がパスを構成する素子やネットのリストを作成する。

【0026】ここまで処理によつていくつかの解析結果表示ができるのでそれらについて説明する。図5はブロックパネルの表示を示している。ブロックパネルには入力したレジスタ転送レベル論理記述中にあるブロック名(Block Name)、そのブロックの上位階層名(Up Block)、そのブロック内の論理記述からゲート論理推定を行つたときの最もクリティカルな論理バスの論理段数(Dansu)、発生した総素子数(Cell No)、面積(Size)、その配置位置(Location)と消費電力(Power)が表示される。面積、ブロック配置位置と消費電力は論理最適化後に計算されたときに表示される。論理段数や素子数についてはこの段階でも表示できるし、論理最適化手段13実行後にも最適化された論理について再計算・再表示される。

【0027】図6はブロックパネル上での階層関係の上下方法を示している。Block Nameの欄の名前をクリックするとUp, Downの表示が出るので、階層を下がるならDownを押せばよい。この例ではkaisouというブロックの下に3つの素子もしくはブロックがあることが分かる。

【0028】図7はブロックパネルとレジスタ転送レベル論理記述表示パネルとの連動を示している。ブロックパネルの素子名をクリックすると、レジスタ転送レベル論理記述表示パネル上で対応する記述がハイライトする。この例では#name2#1という素子がレジスタ転送レベル論理記述上で、kaisouというブロックの記述の中で用いられているnakaというブロックのname2というインスタンスということが示される。また、nakaという記述の先頭にもハイライトが行われている。

【0029】図8はパスパネルを示している。このパネルには論理バスの始点(Source)と終点(Sink)、バス全体のディレイ(Delay)、ゲート論理推定時の論理段数(Dansu)及び論理最適化後の論理段数(Tadan)、バスの種類(Kid)、バスが属するクロックの周期(Cyc)を表示することができる。始点終点間にいくつものバス経路があるときは最も論理段数の大きな経路を表示する。また、各項目についてソートができるようになっており、始点や終点をアルファベット順に並べたり、論理段数やディレイを大きい順に並べたりすることができるので解析結果から

の問題点がすぐに分かる。バスの種類からはそのバスがラッチーラッチ間、ラッチー入出力ピン間、入出力ピン一入出力ピン間なのかがひとめで分かり、バスの種類によって異なるディレイ制約に対する違反がすぐに分かる。

【0030】図9はパスパネルとヒストグラムの連動表示を示す。Histogramのボタンをクリックすると論理段数とその論理段数を持つバスの数が表示される。これによりひとめで論理段数によるバスのディレイの分布が分かる。図10は階層の上下とパスパネルの連動を示している。ブロックパネルで階層を上下し、そのブロック名をクリックするとクリックされたブロック内に含まれるバスが表示される。これにより各ブロック単位に問題点が把握できる。

【0031】図11はパスパネルとレジスタ転送レベル記述表示パネル及びネットパネルとの連動を示している。論理バスの番号をクリックするとそのバス中に含まれる全てのネットがレジスタ転送レベル記述表示パネル上でハイライトする。このハイライトは階層をまたがって、全ての階層で表示される。また、ネットパネルが表示され、バス中に含まれる全てのネット(Net Name)、ネットが含まれるブロック名(Block)、その上位のブロック名(Upper)、そのネットの出力セル(OutputCell)と入力セル(InputCell)、出力セルの論理段数(Dansu)とその出力セルまでの論理段数和(Sum Dansu)、ネットのファンアウト(Fanout)が表示される。このネットパネルによりバス経路の詳細が把握でき、バス経路中のどのネットや素子でディレイの問題が生じているかがすぐに理解できる。図12はネットパネルとレジスタ転送レベル記述表示パネルとの連動を示している。ネットに対応する番号をクリックするとレジスタ転送レベル記述表示パネル上でそのネットとそのネットの出力素子が表示されるので、バス経路をひとつづつ順に追跡することができる。

【0032】ここまでがゲート論理推定手段12が終了したときに表示手段22によって表示できる情報と情報間の連動を示している。この表示手段22によってレジスタ転送レベル記述表示と各評価値の対応が設計者にすぐ分かるようになり、より良いレジスタ転送レベル記述を作り上げるのに効果を持つ。

【0033】次に論理最適化手段13について説明する。まずBDD(Binary Decision Diagram)作成手段51が各出力ピンに関してそのピンに至るまでに関係する全ての入力論理を抽出する。そして冗長回路削除手段52が、0・1どちらを与えても同じ出力を出す部分を探して削除する。そうやって縮約されたBDDを縮約後BDD登録手段53がメモリに格納し、同時に使用可能セルへの置き換え手段54が論理最適化用lib102に格納されている使用可能な素子に置き換える。ここで使用可能な素子の例は2入力NAND、2入力NOR、インバータ、2入力XOR、2入力XNOR等である。論理最適化への指示がディレイ優先の場合にはこの

段階で論理段数計算手段が論理段数を求めて処理を終了する。

【0034】論理最適化への指示が面積優先の場合には、縮約されたBDDに対して、共通部分の縮約手段57が1つの出力ピンへの部分論理の中で他の出力ピンへの部分論理と共通化できる部分の候補を見つけ、実際に共通化できれば共通項を括りだしてさらにBDDを縮約する。最後に縮約後BDD登録手段53が最適化論理を登録する。

【0035】論理最適化が終わるとその情報は、図5のブロックパネルの最大論理段数と素子数の欄、図8のバスパネルの各バスに対する論理最適化後の段数等に反映され、さらに論理最適化で作成された論理はその後の面積やディレイ計算、消費電力計算に使われる。

【0036】次にブロック面積推定手段14について説明する。先に述べた論理最適化手段が終了すると、論理最適化後素子数計算手段61が論理最適化後の素子数を求めて、素子面積を計算する。その素子面積を用いてブロックの面積を推定するのであるが、実際にLSIチップを作るときに使用される配線層の総数によって推定方法が変わることもある。配線層が総数4層以上の場合、一般的に、配線領域を特別に取らなくとも十分全ての配線を行うことができる。この場合は配線領域を推定することなく指定実装率入力手段62が面積計算のための実装率(ブロック面積に対する素子面積の割合)を入力して、面積計算手段63が素子面積合計に実装率を乗じてブロック面積を求める。

【0037】一方、配線層総数が3層以下の場合には配線領域を推定する必要がある。配線層が3層以下の場合は、一般にブロック内のネットの接続を行うのに必要な配線を収納するだけの配線領域を確保しなければならないからである。配線領域を推定するためにはネットの接続を行うのに必要な配線長を求める必要がある。そのためピシ数カウント手段64が配線にかかる指標である素子のピン数を数える。次に総配線長計算手段65が素子数と総ピン数によって必要な配線長を予測する。そして配線長考慮面積推定手段66が、自動配線プログラムが与えられた領域のどのくらいを使える能力があるかというパラメータによってブロック面積を推定する。

【0038】このようにして求めた各ブロックの面積を用いてフロアプラン15を行う。図15は各ブロックをチップ上に配置するための指示方法を示している。ブロックパネルでブロック名をクリックすると、先ほど階層の上下を行った指示パネルが現れ、そこでinformationを選択すると図中にあるサイズとロケーションを入力するパネルが出る。デフォルト値は、サイズはブロック面積推定手段14で見積った値、ロケーションはチップの左下点である。そうすると図中のFloorplanのパネルにあるようにブロックが配置される。

【0039】図16はフロアプラン15で可能な処理を示している。フロアプラン15ではブロックの移動、ブロック

の形状の変更が可能である。移動時には他のブロックとの接続関係を右側の図のネットのような表示で、接続ネットがたくさんある場合には関係の線を太くして表示する。形状変更は見積った面積を保ったままと自由に変更できるモードの2つがある。ブロックの位置関係が決定したら先に解析した論理バスの経路も表示することができる。パネル間の連動に関しては、ブロックパネルでブロックを選択するとフロアプランパネル上でブロックの色が変わったり、その逆にフロアプランパネル上でブロックを選択するとブロックパネルの対応するブロックの色が変わる。論理バスの表示に関してもバスパネルと相互に連動する。

【0040】フロアプラン15によって各ブロックの物理的な形状や位置関係が決まると配線ディレイを含めたバスディレイを計算することができる。バスディレイ計算手段16においてまず全てのネットについてネットディレイを計算する。ネットディレイ計算で考慮する項目を図18に示す。素子に関しては出力ピンの出力抵抗と入力ピンの入力容量を考慮する。配線パターンに関しては線分の容量・抵抗、配線層を変えるときに用いるスルーホールの抵抗を考慮する。

【0041】配線長計算手段71はブロック内、ブロック間の全てのネットについて配線長を計算する。次に配線容量計算手段72がその配線長に各配線層の平均的な容量係数を乗じることによって配線容量を求める。レジスタ転送レベルの段階ではどの配線層に実際の経路が配線されるか分からないので、まず配線長を求め、それに平均容量係数を乗じるのである。この平均容量係数には配線同士が並行して走ったときや交差したときの容量増分が含まれている。容量の考え方について図19に示す。配線容量としては線分と拡散層間の容量であるCs(配線の底面と拡散層との容量)とCf(配線の側面と拡散層との容量)、並行線分間の容量Cp、交差線分間の容量Ccを考えられ、これらを全て考慮した平均容量係数がデフォルトパラメータファイル103に登録してある。次に配線抵抗計算手段73が配線抵抗を求める。抵抗としては線分の長さに比例する配線抵抗と、スルーホールの抵抗を両方考慮する。平均的なスルーホールの数は自動配線プログラムの試行結果から求め、デフォルトパラメータとして登録してある。これらの情報からネットディレイ計算手段74がネットディレイを計算する。

【0042】ネットディレイ計算が終わると論理バス探索手段50によりもう一度バス経路探索を行う。これはブロックの物理的な位置の決定によりネットディレイが精度良く求まり、論理バスのディレイがゲート論理推定時と異なってくるからである。バス経路探索が終了したらクリティカルバス登録手段75がクリティカルバスを登録し、表示手段22によってバスディレイを再表示する。

【0043】つぎに消費電力計算手段18について説明する。消費電力計算は論理シミュレーションの結果を利用

する場合、テストパターンを自動生成して論理設計者のテストパターン作成工数を削減しておいて論理シミュレーションを行いその結果を利用する場合、そして入力ピンに対する信号の変化確率と密度を与えることによって確率的に高速に消費電力を求める場合の3つに分かれれる。

【0044】論理設計者がレジスタ転送レベル記述に対して論理シミュレーションを行なった結果がすでにあるときにはその結果をシミュレーション結果入力手段81が入力する。次にスイッチング回数計算手段82が、ゲート論理推定手段によって推定された初期論理上の各素子に対して、その素子が何回on, offしたかをカウントする。1素子の消費電力は、その素子が動作するときの電圧、その素子の出力ピンにつながる容量、そしてスイッチング回数によって概算できるので、次に配線容量入力手段83がその素子の出力ピンにつながる配線容量と入力ピンの入力容量を合計する。そしてセル電力計算手段84が電圧の2乗・容量・スイッチング回数を乗じて1素子の消費電力を求める。各素子について消費電力が求まると電力集計手段85がブロック全体の消費電力を集計する。

【0045】次に論理シミュレーション結果がない場合について説明する。この場合には確率的に消費電力を計算するか、論理シミュレーションを実行するかのどちらかになるが、論理シミュレーションを実行したい場合にはそのためのテストパターン作成をする必要がある。どちらの場合でも入力ピンの変化確率・密度入力手段87がを入力ピンの変化確率と変化密度を入力する。ここで変化確率とは入力ピンに入る信号の値が1である割合、つまりNクロック周期内に信号が1になる回数の割合を示す。また、変化密度とは信号の値が変化する割合、つまりNクロック周期内に信号が1から0もしくは0から1に変化する回数の割合である。

【0046】入力ピンの変化確率と変化密度が与えられたときのテストパターン作成手段80について図25を用いて説明する。まず、作成ビット数・作成パターン数入力手段95がテストパターンを作成するビット数(LSIの外部入力ピン数)と作成パターン数(0, 1の列数)を入力する。次に乱数発生手段96が作成ビット数分だけ乱数を発生させる。そして、乱数の値が0.5以上ならビットを1に、それ以外ならビットを0にする。この処理によって変化確率が指定されない標準の時のテストパターンが作成できる。指定確率があるときは、0, 1の変更手段97ができたビット列に対して指定確率に応じて0, 1を変更することにより1パターン列が作成される。さらに作成パターン数分これを繰り返すことにより設計者が必要とするテストパターン列を作成することができる。このテストパターンを用いて論理シミュレーションを行ない、その結果を入力することにより消費電力を求める。ただし、この手段によって作成されたテストパターンは

あくまで消費電力計算用のものであり、レジスタ転送レベル記述の動作を検証するものではない。

【0047】最後に入力ピンの変化確率と変化密度から直接消費電力を計算する方法について示す。入力ピンに変化確率・密度が与えられると、確率・密度伝播手段86により、入力ピンにつながる各素子がもっている機能に応じた確率・密度伝播式にその値が代入され、素子の出力ピンに新たな確率・密度が伝播される。これを論理上の全てのパスについて行う。これにより各素子が動作する総回数、つまりスイッチング回数が求められる。その後は論理シミュレーションから電力を求めるのと同じ手段により消費電力が求められる。

【0048】図21には消費電力計算結果の表示方法を示している。ブロック単位の計算結果は図5のブロックパネル内のPowerの欄に表示されるが、その他の表示方法として機能別電力、時刻別電力を表示する。機能別に電力を表示することによって、単にブロック単位で電力が大きいというだけではわかりにくい問題点に対してただちに認識ができる。例えばI/O系の電力が高い場合にはI/O系の論理記述を見直せばよいことがすぐにわかる。また、時刻別に電力を表示することによって、時刻単位でのブロック単位の電力がすぐにわかり、ある時刻では動作していないはずのブロックの電力が高かったりしたときの問題点がただちに認識ができる。図22ではこの解析表示機能を利用して、時刻T1～T2ではブロックDを動作させないようにブロックDへのクロック供給経路にNANDゲートをいれてクロックを供給しないよう回路を構成した例である。

【0049】次に大規模論理評価手段90について説明する。近年は1つのLSIに相当数の論理を搭載することができるので、論理が大規模で階層的に記述されているのが一般的である。大規模なレジスタ転送レベル記述を一挙に解析しようとすると計算機の使用可能メモリを越えてしまう場合があるので、最初に下位階層ブロック数だけ、各ブロック単位に解析し後に全体をまとめて解析を行なう。下位階層ブロックの解析は図1に示した流れと同じである。ただし、フロアプラン15は行なわないのと解析の最後にブロックlib出力手段91によって解析結果を簡略化して出力することが異なる。

【0050】ブロックlibはレジスタ転送レベル記述の解析結果を簡略化して持つが、その簡略化の方法を図24に示す。図24の上図は図1の解析手段によって解析された論理バス経路を示す。Aはバス経路の始点、B, Cは終点を示し、その間の箱は素子を、そしてその中の数字は素子の論理段数もしくは消費電力を示す。バス経路上には多数の素子があるが、素子全てと素子間のネット全てをそのまま表現せずに、1つのバスを1つの素子とその両端のネットで表現する。これによって経路を表現するデータ量が削減される。また、素子の論理段数や消費電力をまとめて1つの素子の値として持つことにより論理バ

ス単位の情報量も少なくすることができる。たとえば箱のなかの数字を素子の論理段数としたとき、AB間の論理段数は2、AC間の論理段数は3であるが一つにまとめた素子に2や3の値を持たせている。

【0051】下位階層ブロックの解析が終了すると最上位の論理記述を入力し、言語構文解析手段11、ゲート論理推定手段12と行なっていく。そして下位階層ブロックの処理を呼び出す記述が現われるとブロックlib入力手段92が解析済み情報であるブロックlib106の内容を入力し、全体解析情報を作る。その全体解析情報に対してフロアプラン15・バスディレイ計算手段16・消費電力計算手段17を行ない、全体解析結果を得る。

【0052】上に述べたようにして1つのレジスタ転送レベル記述の評価を行なうのであるが、面積・ディレイ・消費電力の3つとも最良の値を持つとは限らない。そこで別のアルゴリズムを考え、別のレジスタ転送レベル記述を行なうこともある。その場合は1つのレジスタ転送レベル記述内のハードウェア性能評価値のトレードオフだけでなく、複数のレジスタ転送レベル記述間でのトレードオフ解析が必要になる。図26は複数のレジスタ転送レベル記述間でのトレードオフ解析の様子を示している。本評価システムでは図1の最後に示すこのような記述間の比較がひとめでわかるような方式間評価手段19も備えている。

【0053】

【発明の効果】言語構文解析手段11・ゲート論理推定手段12を持つので、レジスタ転送レベル記述上の全ての論理バスに対してその経路上の詳細な対応全てを表示することができる。これらの手段及び各表示パネル間を連動して表示を行なう表示手段22を備えるので論理記述上の問題点を短時間かつ的確に把握でき、かつレジスタ転送レベル記述の修正時間の短縮及び論理合成回数の削減ができる。

【0054】論理最適化手段13を持つので論理合成システムとは独立に、かつゲートレベル論理を入力することなくLSIのハードウェア性能評価を行なうことができる。ブロック面積推定手段14・フロアプラン15・バスディレイ計算手段16・消費電力計算手段17を持つでLSIの重要なハードウェア性能指標である面積・ディレイ・消費電力の3つを一挙に解析しトレードオフを解析することができる。また、ハードウェア性能評価を短時間で行なうことができるので、何回となくレジスタ転送レベル記述論理を修正して性能向上する工数を削減することができる。フロアプラン15により配線容量を考慮したバスディレイ評価・消費電力評価ができる。

【0055】ブロックlib出力手段91・ブロックlib入力手段92を持つで大規模論理を効率良く解析できる。さらに方式間評価手段19によって複数のレジスタ転送レベル記述に対してその評価値を比較表示する手段を持つのでトレードオフを考慮しながらより良い記述を容易に見

つけだすことができる。

【0056】テストパターン作成手段80を持つので人手によってあらかじめ消費電力計算用のテストパターン作成を行なう必要がなく評価期間を削減することができる。また、確率的に消費電力計算を行なう手段を持つのでテストパターンなしでも消費電力を計算することができる。

【0057】以上の効果を総合して図27に示すような設計期間短縮の効果が得られる。

【図面の簡単な説明】

【図1】本発明のシステム構成。

【図2】本発明のシステムの入出力構成。

【図3】構文解析手段の構成。

【図4】ゲート論理推定手段の構成。

【図5】ブロックパネル表示の例。

【図6】論理階層の上下表示の例

【図7】ブロックパネルとレジスタ転送レベル記述表示パネルの連動の例。

【図8】バスパネル表示の例。

【図9】バスのヒストグラム表示の例。

【図10】論理階層とバスパネルの連動の例。

【図11】バスパネルとレジスタ転送レベル記述表示パネル及びネットパネルの連動の例。

【図12】ネットパネルとレジスタ転送レベル記述表示パネルの連動の例。

【図13】論理最適化手段の構成。

【図14】ブロック面積推定手段の構成。

【図15】ブロックパネルとフロアプランブロック配置の連動の例。

【図16】フロアプランの機能のイメージ図。

【図17】バスディレイ計算手段の構成。

【図18】ネットディレイ計算時の考慮要素の例。

【図19】考慮する配線容量の例。

【図20】消費電力計算手段の構成。

【図21】消費電力解析機能の例。

【図22】消費電力削減効果の例。

【図23】大規模論理評価手段の構成。

【図24】ブロックlibの解析情報の形式の例。

【図25】テストパターン自動発生手段の構成。

【図26】方式間比較の例。

【図27】レジスタ転送レベル記述評価による設計期間短縮の例。

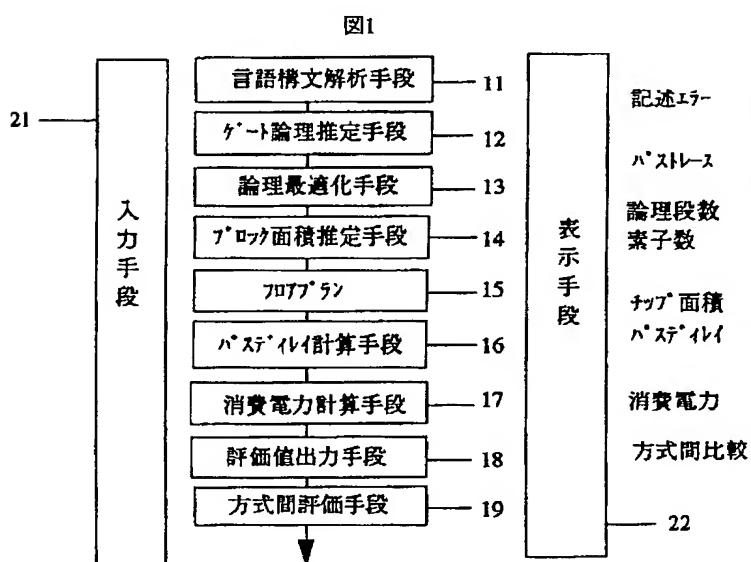
【符号の説明】

11…言語構文解析手段、 12…ゲート論理推定手段、 13…論理最適化手段、 14…ブロック面積推定手段、 15…フロアプラン、 16…バスディレイ計算手段、 17…消費電力計算手段、 18…評価値出力手段、 19…方式間評価手段、 21…入力手段、 22…出力手段、 31…ソース文字列入力手段、 32…ソース行管理手段、 33…単語への分解手段、 34…ファイル管理手段、 35…単語位置管理

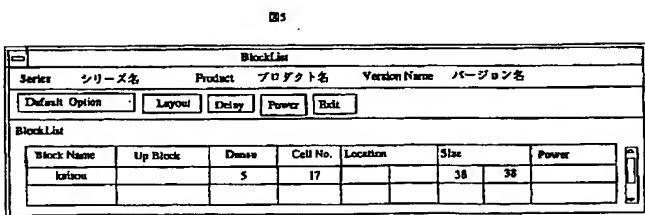
手段、36…単語置換手段、37…単語相関解析手段、41…ネット宣言解析手段、42…初期ネット作成手段、43…単語分類別素子発生手段、44…単語素子間表示位置対応付け手段、45…非明示ネット作成手段、46…ネット接続手段、47…単語ネット間表示位置対応付け手段、48…ネット候補リスト更新手段、49…外部接続ネット作成手段、50…論理パス探索手段、51…BDD作成手段、52…冗長回路削除手段、53…縮約後のBDD登録手段、54…使用可能セルへの置換手段、55…論理段数計算手段、56…縮約後のBDD取出し手段、57…共通部分の縮約手段、61…論理最適化後素子数計算手段、62…指定実装率入力手段、63…面積計算手段、64…ピン数カウント手段、65…総配線長計算手段、66…配線長考慮面積計算手段、71…配線長計算手段、72…配線容量計算手段、73…配線抵抗計算手段、74…ネットディレ

イ計算手段、75…クリティカルパス登録手段、80…テストパターン自動発生手段、81…シミュレーション結果入力手段、82…スイッチング回数計算手段、83…配線容量入力手段、84…セル電力計算手段、85…電力集計手段、86…確率・密度伝播手段、87…入力ピンの変化確率・密度入力手段、90…大規模論理評価手段、91…ブロックlib出力手段、92…ブロックlib入力手段、95…作成ビット・作成パターン数入力手段、96…乱数発生手段、97…0・1の変更手段、101…レジスタ転送レベル論理記述(RTL)、102…論理最適化用lib、103…デフォルトパラメータファイル、104…LEF(Library Exchange Format)、105…PDEF(Physical Design Exchange Format)、106…ブロックib、107…dumpfile(途中結果ファイル)、108…各方式の評価結果ファイル。

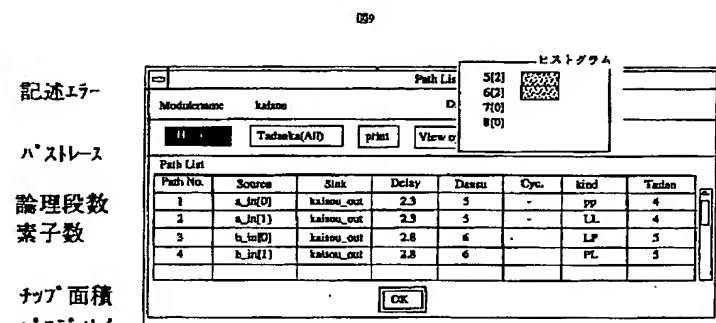
【図1】



【図5】



【図9】



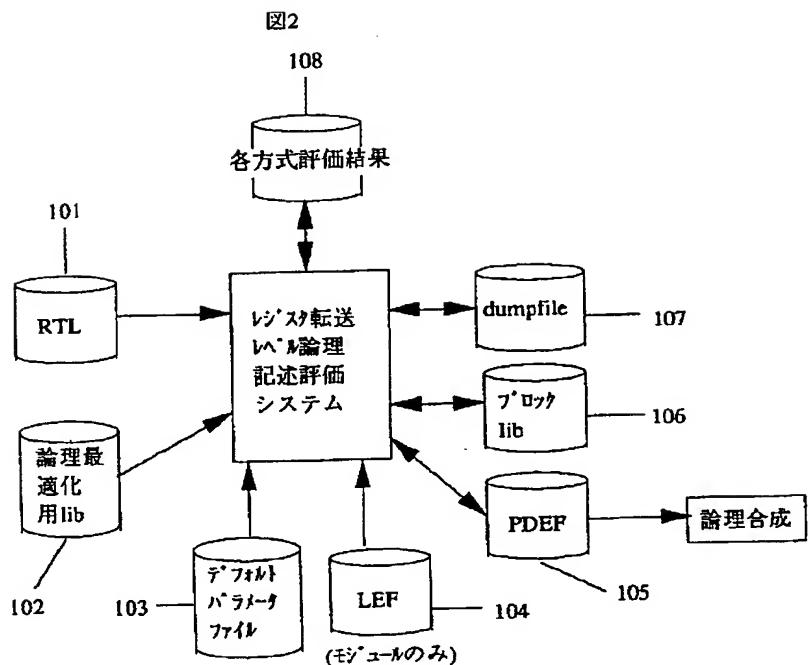
【図7】

```

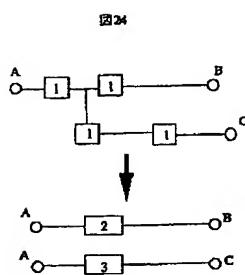
module kalou(a_in, b_in, kalou_out);
    output kalou_out;
    input [1:0] a_in;
    wire [1:0] c_out;
    naka naka1(c_out[0], a_in[0], b_in[0]);
    naka naka2(c_out[1], a_in[1], b_in[1]);
    assign kalou_out = c_out[0] + c_out[1];
endmodule;
module naka(a_out, naka_in1, naka_in2);
    output a_out;
    input naka_in1, naka_in2;
    wire d_out1, d_out2;
    naka (d_out1, naka_in1);
    naka (d_out2, naka_in2);
    assign a_out = d_out1 + d_out2;
endmodule;

```

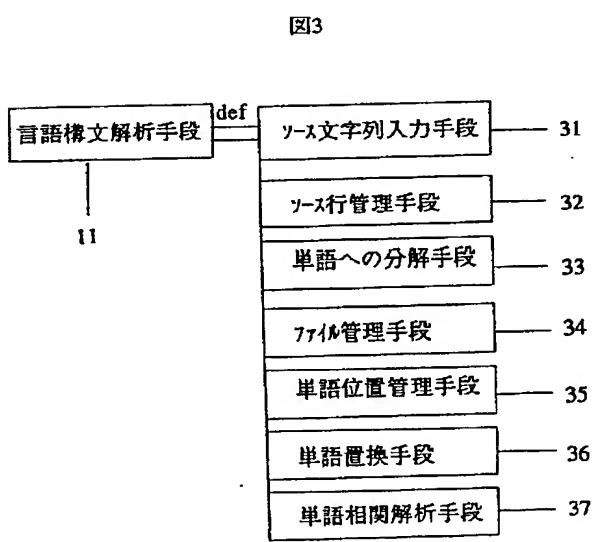
【図2】



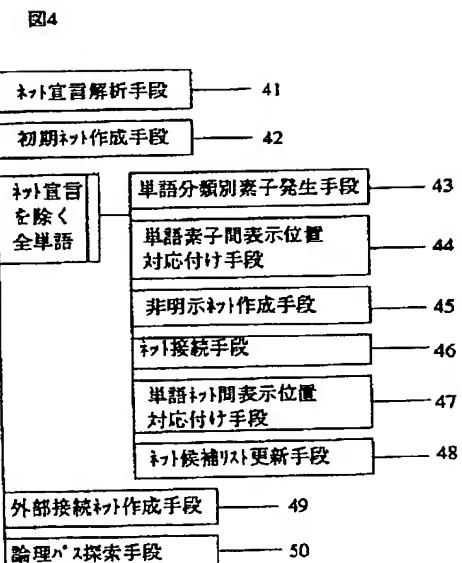
【図24】



【図3】



【図4】



【図6】

図6

Block Name	Up Block	Status	Call No.	Location	Size	Power
kaishou	Up	5	17		38	38
+2@INT#1						
#name1#	information	3	9		23	23
#name1#	kaishou	3	9		23	23

【図10】

■10

図8

Path No.	Source	Sink	Delay	Status	Cyc.	kind	Tadan
1	a_in[0]	kaishou_out	2.3	5	-	PP	4
2	a_in[1]	kaishou_out	2.3	5	-	LL	4
3	b_in[0]	kaishou_out	2.8	6	-	LP	5
4	b_in[1]	kaishou_out	2.8	6	-	PL	5

【図11】

■11

図12

Path No.	Source	Sink	Delay	Status	Cyc.	kind	Tadan
1	maka_in1	maka_out	L6	3	-	PP	2
2	maka_in2	maka_out	L8	4	-	PP	3

【図12】

■12

図11

```

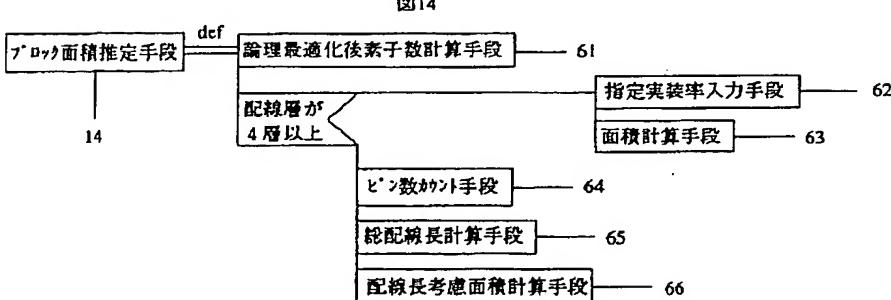
module kaishou(a_in, b_in, kaishou_out);
output kaishou_out;
input [1:0] a_in, b_in;
wire [1:0] c_out;
naka name1 (.c_out[0], a_in[0], b_in[0]);
naka name2 (.c_out[1], a_in[1], b_in[1]);
assign kaishou_out = name1 + name2;
endmodule
module naka (naka_out, naka_in, naka_in2);
output naka_out;
input naka_in, naka_in2;
wire d_out1, d_out2;
site (d_out1, naka_in);
site (d_out2, naka_in2);
assign naka_out = d_out1 + d_out2;
endmodule

```

No	Net Name	Block	Upper	OutputCell	Amount	Status	Sum Dm	InputCell
1	a_in[0]	kaishou		Ext Pin	1	0	0	#name1#
2	naka_in1	name1	kaishou	Ext Pin	1	0	0	#name1#
3	naka_in	naka	name1	Ext Pin	1	0	0	+1@INT#1
4	naka_out	naka	name1	+1@INT#1	1	1	1	#name1#
5	d_out1	name1	kaishou	site#1	1	1	1	+2@INT#1
6	naka_out	name1	kaishou	+2@INT#1	1	2	3	#name1#
7	c_out[0]	kaishou		site#1	1	3	3	+2@INT#1
8	kaishou_out	kaishou		+2@INT#1	1	2	5	Ext Pin

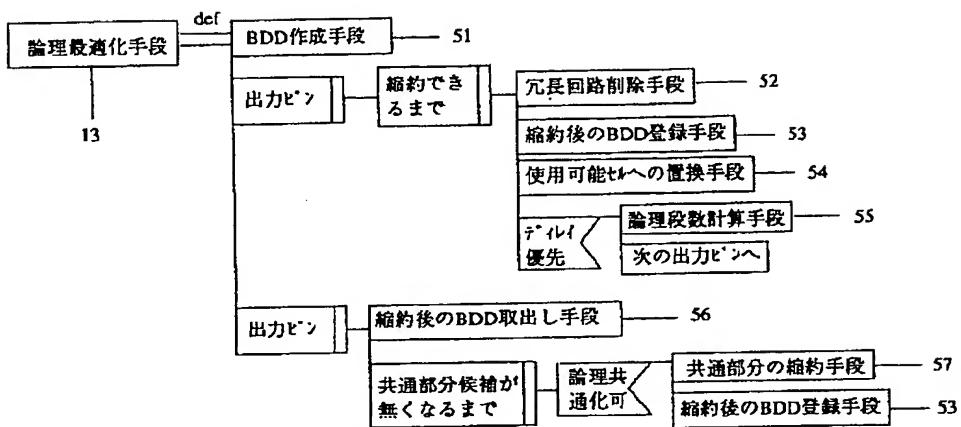
【図14】

図14



【図13】

図13

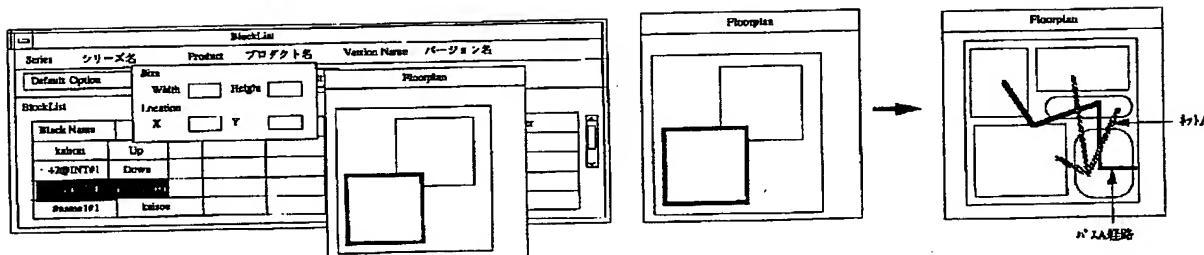


【図15】

【図16】

図15

図16

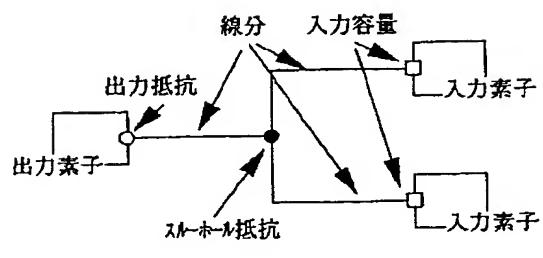
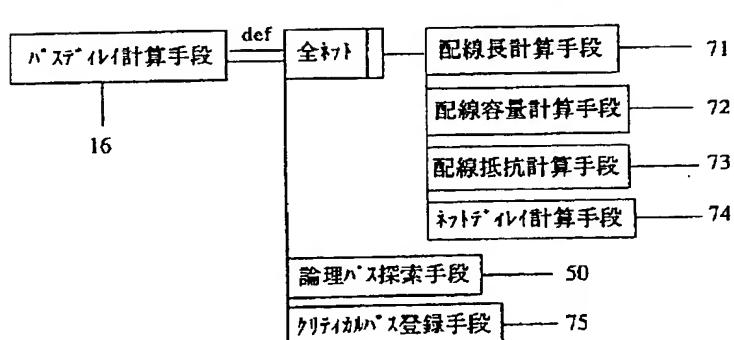


【図18】

【図17】

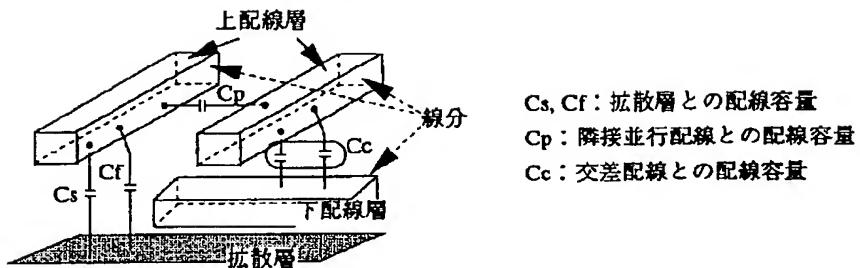
図18

図17



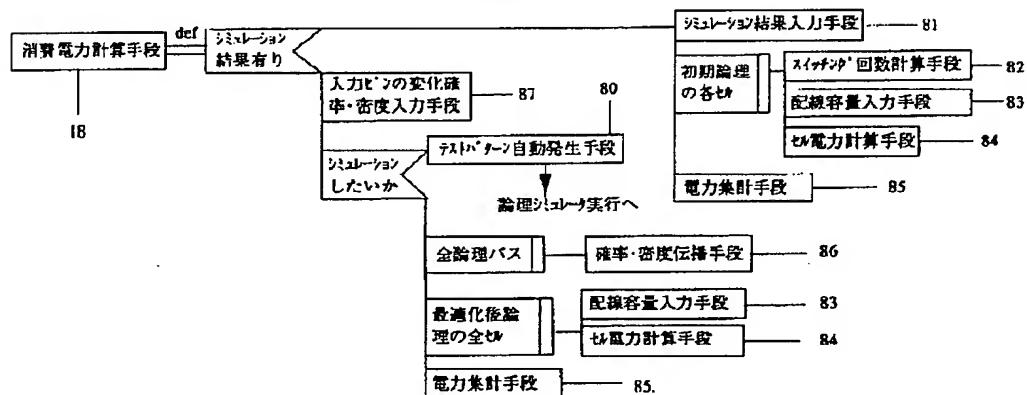
【図19】

図19



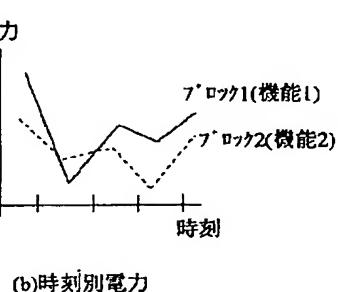
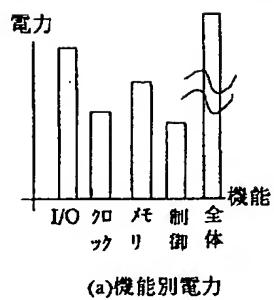
【図20】

図20



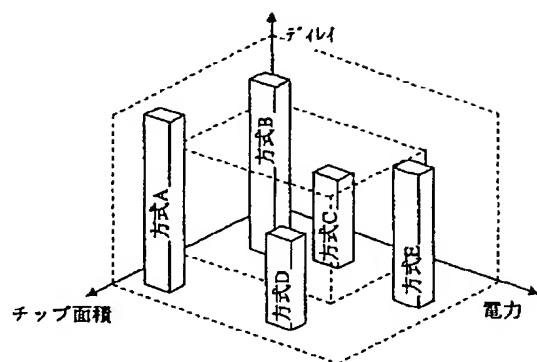
【図21】

図21



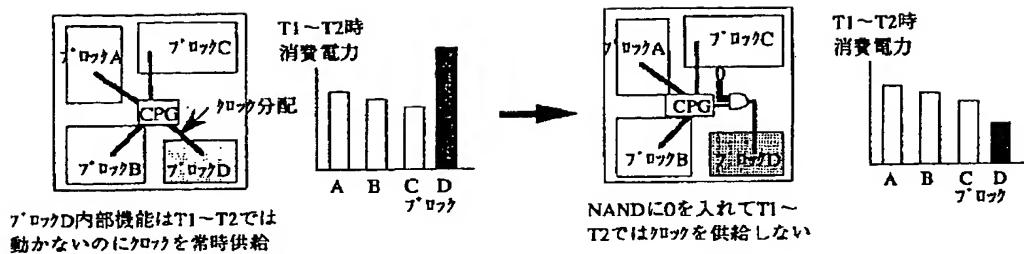
【図26】

図26



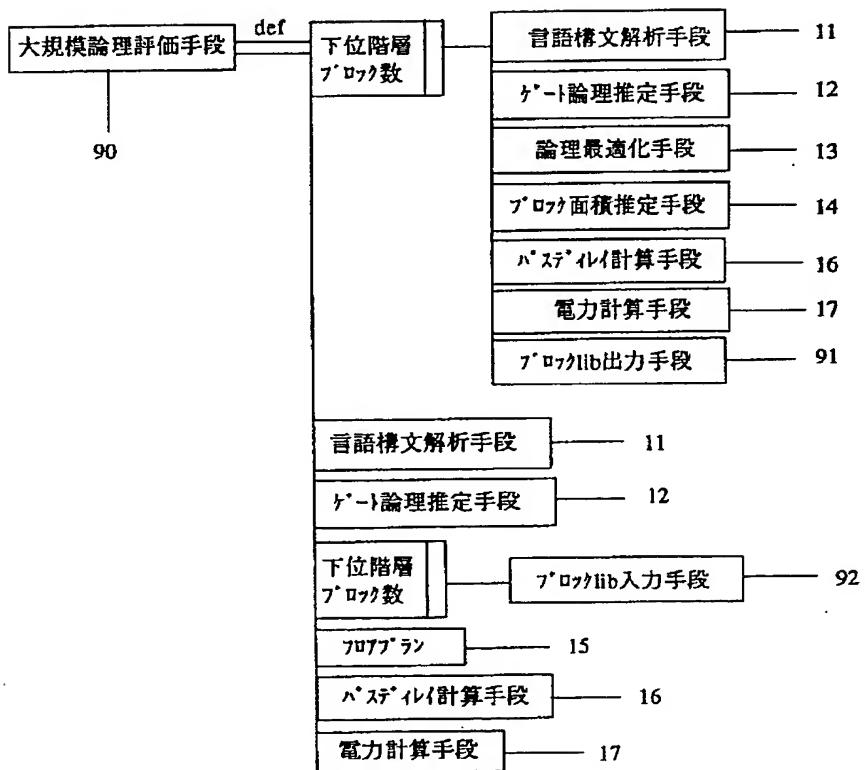
【図22】

図22

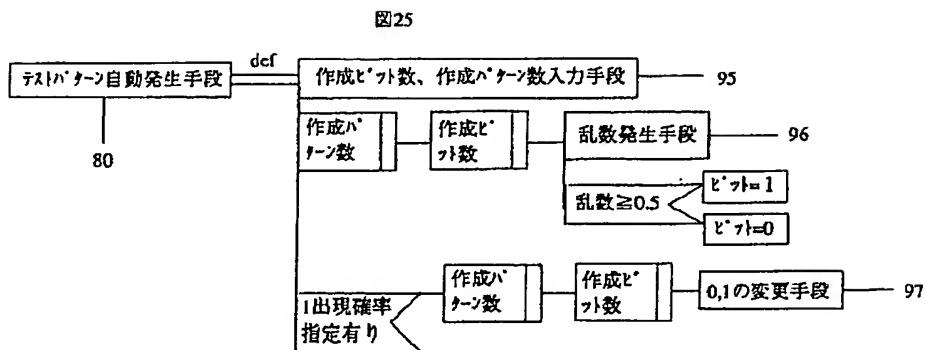


【図23】

図23



[図25]



[図27]

図27

